

Prolegomenon to a Media Theory of Machine Learning: Compute-Computing and Compute-Computed

DAVID M. BERRY

University of Sussex, UK

PREPRINT FROM JOURNAL OF MEDIA THEORY

<http://mediatheoryjournal.org>

In this paper, I outline a groundwork for a media theory of machine learning by introducing two new concepts, *compute-computing* and *compute-computed*, and a framework for their interaction.^[1] *Compute-computing* (computing as generative) is here understood as the “active” learning component of a system, whereas *compute-computed* (computing as generated) is understood as the “passive”, coded, imprinted or inscribed aspect of a system. I introduce these two concepts to help us to think through the specificity of algorithmic systems that are more than just the operative, sequential or parallel systems of computational processing to which we have become accustomed. Indeed, in the case of machine learning systems, these systems have the capacity to be self-positing in the sense of generating models and data structures that internalise certain pattern characteristics of data, without the requirement that they are translated into formal data structures by a human programmer.^[2] That is, they are able to capture the abstract form of data input into the system, identify key characteristics, frames or patterns, and store this for comparison and classification of other data streams or objects.

In a sense, these systems could be said to have an additional agency which is the ability to create new algorithms, as *compute-computing*; that is, that they can construct a model of a “world” of data and functions to transform them. Due to

limitations of space, I can only give the broad outlines of the theoretical and conceptual work that needs to be undertaken to think through this new computational form and its implications. Nonetheless, I do want to point towards the possible future directions for thinking about machine learning that this preliminary work suggests. Consequently, in this paper I am forced to bracket out the broader societal and political economic implications – which are of course, substantial and estimated at \$26-39 billion in investment in machine learning in 2016 alone (Bughin et al., 2017) – in order to concentrate on a new framework for thinking about machine learning, but also as a contribution towards critiquing it. In particular, I want to think about machine learning in terms of its capacity for self-writing, or automatic model-building, and the problematics for thinking about the complexity of code, software and algorithms when the “code” is, in some sense, wrapped again inside another level of complexity. Machine learning appears as a “riddle, wrapped in a mystery, inside an enigma”, particularly to those outside the field, and this morphology of obscurity and complexity requires theoretical and empirical unpacking.^[3] By presenting a different conceptual model for thinking about machine learning, we can begin the critical work of understanding what is happening beneath the surface of these new computational forms and how their deployment matters.

The difficulty of researching algorithms and software have been the focus of a number of scholarly works (for example: Berry, 2011; Berry and Fagerjord, 2017; Chun, 2006; 2010; 2011; Fuller, 2008; Manovich, 2001; Marino, 2006; Wardrip-Fruin, 2009) but here, by concentrating on machine learning as a specific problematic in relation to software more generally, I want to pull attention towards the particular issues for a media theory of machine learning. There are already some useful examples of scholarly work that are thinking about machine learning, but these are usually overly general (Alpaydin, 2016; Domingos, 2017), focused on issues of black-boxing or technical classification (see Burrell, 2016; Domingos, 2012) or are popular texts concerned with broader themes (Carr, 2016; Ford, 2016; Kelly, 2017; McAfee and Brynjolfsson, 2017). In contrast, here I develop what we might call a theoretical and philosophical prolegomenon for a new set of concepts for thinking about machine learning structures and processes.

In the first section, I want to look at the specificity of machine learning in relation to the larger field of artificial intelligence research. In the second section, I introduce my conceptual framework and explain how it is linked to the work of Baruch Spinoza. Finally, in the last section, I explore how this conceptual framework provides a means for discussing machine learning in relation to problematics raised for media theory, digital humanities and social theory more generally. As previously noted, this paper, by virtue of its restricted length, will be limited in offering a broad overview of very complex subjects; nonetheless, it is hoped that this helps to concentrate the discussion around the theoretical concepts I want to outline. [\[4\]](#)

1.

In 1959, Arthur Samuel is claimed by many who work in the area to have defined machine learning as a “field of study that gives computers the ability to learn without being explicitly programmed”. I say claimed, because although widely cited in the literature to Samuel’s work (1959), the phrase does not actually appear in his article. Nonetheless, this definition is an accepted (and repeated) origin point in the field and often used to show how machine learning is particularly geared towards the self-learning capacity of a machine and how it differs from artificial intelligence, that is, the application of computation to symbolic tasks that are usually undertaken through human cognition.^{[\[5\]](#)} In 1997, Mitchell updated this definition to describe machine learning as “a computer program [that] is said to learn from experience E with respect to some class of tasks T and performance measure P , if its performance at tasks in T , as measured by P , improves with experience E ” (Mitchell, 1997). In essence, he argues that it is the performance indicator that is crucial for the development of the “learning” capacity, that is the ability to undertake the processing work within a shorter period of time. We can only note here that this notion of “learning” is very specific and technical in its deployment, and relates to the ability to undertake skills or tasks, not to wider humanistic connotations of learning as understanding, interpretation, etc. Indeed, it is this focus on specific domain

problems that is said to delimit machine learning in relation to the wider knowledge problems of general artificial intelligence.

Machine learning as a specific area of artificial intelligence research has received a great deal of interest, not only from academic researchers and corporations, but also in terms of the public sphere through the media since 2010 (see Donnelly, 2017; *Economist*, 2017; Lewis-Kraus, 2016; Tufekci, 2014). Partly, this has been due to changes in hardware and software capacities that enable some of the promises of artificial intelligence to be realised across the entire landscape of media ecology. The turn to machine learning has also been driven by the limited capacities within disciplines to cope with an ever-growing mountain of digital data, so-called Big Data, combined with a political economy that sees huge economic potential in mining this data for insights and profit. As Burrell explains, “machine learning algorithms are used as powerful generalizers and predictors. Since the accuracy of these algorithms is known to improve with greater quantities of data to train on, the growing availability of such data in recent years has brought renewed interest to these algorithms” (Burrell, 2016: 5). It is certainly the case that machine learning is finding its way into a myriad of devices, from cloud computing centres, to translation services, televisions, phones and talking assistants. Indeed, the deployment of machine learning has increasingly begun to resemble other kinds of computational services, with a notional layering of abstractions available as code libraries and application programming interfaces (APIs), and also as services available from third parties.

To create systems in this way is to already begin to reveal the depth model that is implicit in computational layers, often wrapped inside each other. Whilst the notion of layering in computational systems is very common (see Berry, 2014: 58; Kitchin, 2016: 20), this is also very much the logic of producing a “black box” that can handle machine learning processing with a simplified interface for inputs and outputs (Berry, 2011: 15-16). However, if we move away from the external perspective on machine learning and focus on its internal system structure, here it is interesting to observe the way in which machine learning is structured rather like an onion, with the outside

layers, usually programmed in a conventional computer programming language, creating an internal software machine that constructs abstractions of data that can be created, linked, programmed and weighted in a number of important ways. Machine learning algorithms have three main aspects which need to be implemented in the development of machine learning systems. These are: (1) *Knowledge Representation*: Machine learning algorithms implement a model of knowledge, using knowledge representations such as decision trees, sets of rules, instances, graphical models, neural networks, support vector machines, or model ensembles; (2) *Evaluation*: Machine learning systems are trained to become classifying systems using inductive learning techniques, and are evaluated through techniques such as accuracy, prediction and recall, squared error, likelihood, posterior probability, cost, margin, and entropy k-L divergence; and (3) *Optimisation*: the algorithms are optimized using techniques such as combinatorial optimization, convex optimization, and constrained optimization.

Most machine learning systems use a model of inductive cognition to produce classifications. Here, induction is understood as “the process of inferring general rules from specific data” (Mooney, 2000: 1). Machine learning can be organised using *supervised learning* with training data, *unsupervised learning* with clustering techniques, *semi-supervised learning* with a smaller amount of training data, or through *reinforcement learning*, whereby feedback into the network reinforces internal structures based on the success of its output. In all cases, the aim is that the system learns to create the function that transforms the input data into an output, to create so called *local generalisation* as opposed to *abstract generalisation*. The output can take the form of *classificatory systems*, where the function creates discrete outputs, *regression systems*, where the function is continuous, and *probability estimation systems*, where the output is a probability value.

For example, where the knowledge representation is a neural network, a “connectionist” system is constructed. In the case of neural networks, a different paradigm for computing is introduced based on processing/memory abstraction that is inspired by the “parallel architecture of animal brains”. The neural net systems work by taking a given input A and translating it into B through intermediate,

sometimes called hidden, layers of neural nets. Traditional computational systems are usually procedural (or imperative); a program starts at the first line of code, executes it, and goes on to the next, following instructions in a somewhat linear fashion.^[6] A true neural network does not necessarily follow a linear path. Rather, information is processed collectively, in parallel throughout a network of nodes (the nodes, in this case, being neurons, or small program units connected together with weights between them). By using techniques such as feed-forward (i.e. no loops in processing) and back-propagation (allowing the output to be weighted back into the network to correct anomalies), these systems can become better at pattern recognition and classification.

The actual internals to the functioning of machine learning might be thought of as analogous to the notion of “sandboxing”, whereby the machine learning model is contained within another structure of code. Sandboxing is a technique used in computing to separate application code into differing levels of access and control so that different security levels can be applied and only the appropriate level of access is granted to the application. The key aim is to prevent unauthorised access to computer resources, but it can also be used to normalise the computational architecture so that the same code can run on different systems, as was done with the Java language, for example. In the case of neural nets, it is not computer security that is the issue as such; rather, it is the creation of an abstract machine that models in software the operation of a simplified notion of neural cells, and which can encode and store a functional transformation within a network data structure. With machine learning, one could say that a different form of “sandboxing” is being undertaken, whereby the “learning” or “training” processes are required to generate an algorithmic model embedded in an abstract machine. This in turn is constructed from software neural networks to generate a function for analysing data inputs and computed outputs (whether classification, regression or probabilistic). This structure allows the network to learn to identify similarities, and as such move from the training data to completely novel data that it can pattern match based on the training data it has seen before. Thus the “fundamental goal of machine learning is to generalize beyond the examples in the training set” (Domingos, 2012). That is, machine

learning is essentially an inductive process based on the original empirical training data fed into the network inputs and carefully reinforced so that the network pattern matching achieves the desired aims.

These systems have a broad range of uses, but some include: *pattern recognition*, with examples such as facial recognition, optical character recognition, etc.; *time series prediction*, so that machine learning can be used to make predictions; *signal processing*, so that machine learning can be trained to process an audio signal and filter it appropriately; *control*, so that machine learning can be used to manage steering decisions of physical vehicles; *soft sensors*, so that analysing a collection of many measurements can be abstracted into one machine learning algorithm by processing input data from many individual sensors and evaluate them as a whole; and lastly for *anomaly detection*, so that the machine learns to recognize patterns, and can also be trained to raise an alert when something is anomalous. These various use cases have endeared machine learning to a number of real-world systems, from financial marketing forecasting, fraud detection and identification systems to anti-terrorism surveillance.

2

Now I want to change register and turn to conceptualising the underlying structure in machine learning, by briefly thinking through the work of the philosopher, Baruch Spinoza (1632-1677).^[7] Whilst there is not sufficient space to give a deep outline of Spinoza's philosophy, what I want to bring forward is his development of the medieval notion of *Natura naturans* (Nature naturing), particularly in relation to Spinoza's notions of *Natura naturans* and *Natura naturata*. The notions of *Natura naturans* and *Natura naturata* can be traced back to ancient Greek philosophy and through Augustine and Scotus Eriugena as a distinction between between God and world. Spinoza uses these concepts but with the intention to introduce a distinction between the part of nature that he argues is generative, and related to a traditional notion of a creator God, that is *Natura naturans*, and the part of nature that is the result of an act of creation, *Natura naturata* (see Demasio, 2003: 329). Thus, for

Spinoza there are two sides of Nature. There is the active, productive aspect of the universe – what Spinoza calls Substance and its attributes, and from which all else is derived and which Spinoza calls *Natura naturans* – ‘naturing Nature’. There is also the other aspect of the universe is that which is produced and sustained by this active aspect, which Spinoza calls *Natura naturata*, or ‘natured Nature’ (see Nadler 2001: 100).

Spinoza uses these terms in *Ethics* (1, xxix) where he explains that “by *Natura naturans* we must understand what is in itself and is conceived through itself, or such attributes of substance as express an eternal and infinite essence, that is ... God, insofar as he is considered as a free cause.” That is, that *Natura naturans* is productive of an objective *Natura naturata*. In other words, there is “causal and epistemic dependence of all things upon God” (as *Natura naturans*) (Nadler, 2001: 100). Whereas he argued that “by *Natura naturata* I understand whatever follows from the necessity of God’s nature, or from God’s attributes, that is, all the modes of God’s attributes insofar as they are considered as things which are in God, and can neither be nor be conceived without God.” So, *Natura naturans* has traditionally designated God, insofar as he is understood as the creator and principle of all action, while *Natura naturans* is understood as the totality of beings and laws he has created (see Hadot, 1995: 262). This construction, as Hadot notes, has had a lot of interest from artists for conceptualising their practice, such as Klee who argued that “*Natura naturans* is more important to the painter than *Natura naturata*” (Hadot, 1995: 255). So, Spinoza understands *Natura naturans* as ‘Nature’ – a creative potency-in-act as God. This is Nature creating itself, nature as “naturing itself”. In contrast, he argues, *Natura naturata* is a determinate totality of determinate being, it having received a form – that is, nature ‘natured’.

3

This distinction can be understood as being of a constitutive (*Natura naturans*) and an operational (*Natura naturata*) form and it is this constitutive and operative distinction that I want to argue is helpful for thinking through machine learning. To

this I want to introduce these two new concepts. (1) *Compute-computing*, understood as a generating level of activity in the machine learning system, which can be understood as “active” and analogous to Spinoza’s notion of *Natura naturans*, in as much as it creates the conditions for the neural network. That is, it forms a constitutional level in the computational structure which creates the conditions of possibility for the network as such. This is in operation mostly in the training or learning phase of the machine learning system. This is analogous to what Spinoza understands as the constitutional structure and patterns, what he calls the “attributes” of *Natura naturans*. And the second concept of (2) *compute-computed*, understood as that which is or has been generated and as such is “passive”, as in Spinoza’s notion of *Natura naturata*. This forms the operational level of the machine learning system, what Spinoza understands as the operational “modes” of *Natura naturata*. So here, passive does not mean inactive; rather it points to the performative aspect of the *compute-computed*, that which has been produced and which can be made operative in relation to the function of pattern matching by the network. This, then, is the network following training with data, and which cannot, of itself, produce another network. With the notion of *Natura naturans*, Spinoza and other philosophers pointed towards the fecundity of Nature (and by extension God) in its creative, dynamic capacity. Similarly, here I want to point to the creative potentiality offered by *compute-computing* for generating multiple modes of *compute-computed*.

We might note that there are a number of different machine learning algorithms. We can think of these as multiple attributes of *compute-computing*. Examples include decision tree learning, association rule learning, artificial neural networks, deep learning, inductive logic programming, support vector machines, clustering, Bayesian networks, reinforcement learning, representation learning, similarity and metric learning, sparse dictionary learning, genetic algorithms, rule-based machine learning, and learning classifier systems. Broadly speaking these algorithms are generative and allow a machine to learn using a learning data set so that it can work with new, unseen data. The idea is that through the learning process the algorithm is

able to create a *compute-computed* which enables some form of prediction or pattern-matching related to new presented data.^[8]

Now I want to change register again and think more broadly about the wider use of machine learning techniques in examples which will be more familiar. I will briefly introduce deep learning algorithms, particularly those which use convolutional neural networks (CNNs). I focus on these because CNNs have application in natural language processing but also in image and video recognition and, therefore, applicability in relation to digital humanities projects.^[9] The wider public are becoming familiar with the generative capacities of so-called deep convolutional networks, such as DeepDream, due to its appeal to popular culture in the generation of seemingly hallucinated images that are dreamlike in form. DeepDream is a system created by Google using a CNN to find and enhance patterns in images via algorithmic pareidolia. The software is designed to detect faces and other patterns in images, with the aim of automatically classifying images. However, once trained, the network can also be run in reverse, being asked to adjust the original image slightly so that a given *compute-computed* (e.g. the image for faces or certain animals) is placed back in the original image.

CNNs work by modelling animal visual perception, and can therefore be applied to visual recognition automation. They are made up of multiple layers of individual software sensory neurons (so-called receptive fields, which are made up of clusters of these neurons). The word “convolution” comes from its use to describe a mathematical operation on two functions which produces a third function. The new function is a modified version of one of the original functions. For image analysis, convolutional filtering plays an important role in many important algorithms; for example, in edge detection, sharpening an image and adding blurring. Convolutional neural networks cascade convolution functions to create high-definition and detailed image analysis. They can also be used to identify and analyse textual inputs, and can recognise different letter forms, typefaces, characters, etc. and generate visualisations based on these contents. One example of their use in a digital humanities project is in Patricia Fumerton’s work (UCSB) in the English Broadside Ballad Archive.^[10] Here, CNNs are used to process and classify woodblock images

and text automatically. The use of machine learning creates a pattern library for finding like woodblocks but also opens up the possibility of discovery of new links between the woodblocks.

Another example of these systems is recurrent neural network (RNN), a class of artificial neural network where connections between networks form a directed cycle. Unlike feedforward neural networks, RNNs can use their internal memory to process arbitrary sequences of inputs. This makes them applicable to tasks such as unsegmented connected handwriting recognition or speech recognition. One such RNN, long short-term memory (LSTM), is a recurrent network that excels at remembering values for either long or short durations of time. Indeed, it is for these reasons that they are used by Google for speech recognition on the smartphone, for the smart assistant Allo, and for Google Translate. Apple also uses LSTM machine learning for the “Quicktype” function on the iPhone and for Siri, and Amazon similarly uses LSTM for Amazon Alexa. These machine learning supported systems are very much becoming more evident in everyday life, and I hope that, by showing these links between what seem like complex and esoteric paradigms in computing and their growing importance as elements of mediation and experience in everyday use of phones, computers and technology, we can see the urgency for media theoretical work in this area.

This short paper can only give a very brief introduction to these questions and to the theoretical work I am developing in this area. With machine learning we have the broad outlines of a new computational paradigm which is likely to have a major impact on the kinds of media systems the public uses over the next decade or so. As I have outlined in this paper, this is a complex technical field but needs to be addressed urgently. I have sought to rethink the technical issues at play by rearticulating their major contours through Spinoza’s concepts of *Natura naturans* and *Natura naturata* to develop the analogous concepts of *compute-computing* and *compute-computed*. These bring to the fore the major advances for thinking about computation in terms of its generative and pattern-matching capacities in recent technical work. This paper has only given broad outlines and examples of

this new and rapidly growing field and much work remains to be done. Indeed, if we have only just started asking questions about the medium specificity of algorithms and software, then it is clear that the added complexity of machine learning is going to challenge media theory into engaging with new forms of computation which have important consequences for human attention, reading, learning, and instrumentality more widely.

References

- Alpaydin, E. (2016) *Machine Learning: The New AI*. Cambridge, MA: The MIT Press.
- Berry, D. M. (2011) *The Philosophy of Software*, Basingstoke: Palgrave Macmillan.
- Berry, D. M. (2014) *Critical Theory and the Digital*, New York: Bloomsbury.
- Berry, D. M. (2017) *Digital Humanities: Knowledge and Critique in a Digital Age*, Cambridge: Polity.
- Bughin, J., Hazan, E., Ramaswamy, S., Chui, M., Allas, T., Dahlström, P., Henke, N. and Trench, M. (2017) *Artificial Intelligence: The Next Digital Frontier?*, discussion paper, McKinsey, <http://www.mckinsey.com/mgi>
- Burrell, J. (2016) 'How the machine 'thinks': Understanding opacity in machine learning algorithms', *Big Data & Society*, January–June 2016: 1–12.
- Carr, N. (2016) *The Glass Cage: Who Needs Humans Anyway*, London: Vintage.
- Chun, W. H. K. (2006) *Control and Freedom: Power and Paranoia in the Age of Fiber Optics*, Cambridge: MIT Press.
- Chun, W. H. K. (2010) 'On Software: Or the Persistence of Visual Knowledge', *Grey Room*, No. 18 (Winter, 2004), pp. 26-51.
- Chun, W. H. K. (2011) *Programmed Visions: Software and Memory*, Cambridge: MIT Press.
- Demasio, A. R. (2003) *Looking for Spinoza: Joy, Sorrow, and the Feeling Brain*, London: William Heinemann.
- Domingos, P. (2012) 'A few useful things to know about machine learning', *Communications of the ACM*, 55(10): 78.
- Domingos, P. (2017) *The Master Algorithm: How the Quest for the Ultimate Learning Machine Will Remake Our World*, London: Penguin.

- Donnelly, P. (2017) 'Why rage against the machines when we could be friends?', *The Guardian*: <https://www.theguardian.com/commentisfree/2017/apr/29/artificial-intelligence-workplace-future-friends>
- Economist* (2017) 'Machine-learning promises to shake up large swathes of finance', *The Economist*: <https://www.economist.com/news/finance-and-economics/21722685-fields-trading-credit-assessment-fraud-prevention-machine-learning>
- Ford, M. (2016) *The Rise of the Robots: Technology and the Threat of Mass Unemployment*, Oneworld Publications.
- Fuller, M. (2008) *Software Studies \ A Lexicon*, London: MIT Press.
- Hadot, P. (1995) *Philosophy as a Way of Life: Spiritual Exercises from Socrates to Foucault*, London: Wiley-Blackwell.
- Lewis-Kraus, G. (2016) 'The Great A.I. Awakening', *The New York Times*: <https://www.nytimes.com/2016/12/14/magazine/the-great-ai-awakening.html>
- Kelly, K. (2017) *The Inevitable: Understanding the 12 Technological Forces That Will Shape Our Future*, London: Penguin.
- Kitchin, R. (2016) 'From a single line of code to an entire city: Reframing the Conceptual Terrain of Code/Space', in Kitchin, R. and Perng, S-Y. (eds.) *Code and the City*. Routledge, London. pp. 15-26.
- Manovich, L. (2001) 'The Language of New Media'. London: MIT Press.
- Marino, M. C. (2006) 'Critical Code Studies', *Electronic Book Review*, accessed 16 Sept 2011: <http://www.electronicbookreview.com/thread/electropoetics/codology>
- McAfee, A. and Brynjolfsson, E. (2017) *Machine, Platform, Crowd: Harnessing the Digital Revolution*, London: W. W. Norton & Company.
- Mitchell, T. M. (1997) *Machine Learning*, London: McGraw-Hill.
- Mooney, R. J. (2000) 'Integrating Abduction and Induction in Machine Learning', in *Abduction and Induction*, P. Flach and A. Kakas (eds.), Kluwer Academic Publishers, pp.181-191.

- Nadler, S. (2001) *Spinoza's Heresy: Immortality and the Jewish Mind*, Oxford: Oxford University Press.
- Samuel, A. L. (1959) 'Some Studies in Machine Learning Using the Game of Checkers', *IBM Journal of Research and Development*, Volume:3, Issue: 3: 210
- Tufekci Z (2014) 'The year we get creeped out by the algorithms'. Available at: <http://www.niemanlab.org/2014/12/the-year-we-get-creeped-out-by-algorithms/>
- Wardrip-Fruin, N. (2009) *Expressive Processing: Digital Fictions, Computer Games, and Software Studies*, MIT Press.

Notes

- [1] There are also interesting resonances of Heidegger's notion of things thinging.
- [2] Many of these machine learning systems still require handcrafting by "engineer-artisans" who are required to optimise the networks internal to these systems.
- [3] Winston Churchill in a radio broadcast in October 1939 declared, "I cannot forecast to you the action of Russia. It is a riddle, wrapped in a mystery, inside an enigma; but perhaps there is a key. That key is Russian national interest."
- [4] Journals that engage with theoretical and philosophical concepts and ideas, such as *Media Theory*, are important sites for the exploration and explication of what we might call theory-work, and as such crucial to the development of the field of media philosophy.
- [5] This definition is contested, of course, and many scholars and practitioners think of machine learning as a subset of the wider field of artificial intelligence. The term *artificial intelligence* itself was coined in 1956 at Dartmouth College at the Dartmouth Summer Research Project on Artificial Intelligence by John McCarthy, then an Assistant Professor of Mathematics.
- [6] Object oriented programming and related approaches extend these techniques but continue to require human understanding and programming of its linear operation, albeit in this case distributed over a system of software objects.

[7] It might be noted that there is an interesting tension in using the work of a rationalist such as Spinoza for thinking through the empiricism of machine learning, particularly in light of the emphasis placed on the inductive model of reasoning machine learning relies upon. In this paper, there isn't space to develop this issue and think through the tension in any depth, but I look to engage with this question in a later paper. Many thanks for the discussion by participants at the Visualisierungsprozesse in den Humanities conference which took place in Zürich, Switzerland, from 17-19 July 2017, where this issue and others were raised.

[8] An important area for digital humanities is that of topic modelling, which gives a good example of this distinction between the constitutive (*compute-computing*) and the operative (*compute-computed*) in its use and deployment. Both the discriminative and generative machine learning forms of topic modelling can be helpfully understood using these concepts.

[9] There are also a number of open source projects available to use off-the-shelf: Caffe, DeepLearning4j, DeepLearning-hs, neon, TensorFlow, Theano, and Torch.

[10] English Broadside Ballad Archive: <http://ebba.english.ucsb.edu>